UEFA Champions League Data Analysis Project

Alex Eidt and Pranav Natarajan

Research Question

The UEFA CHAMPIONS LEAGUE DATASET, SUBSET from 2008-2015

Based on past performances in the Champions League, which team is most likely to win the competition?

Subsidiary Research questions(Will provide data to get to the machine learning step):
Compare teams based on their record against other teams.

1. What are the win percentages of all teams in the competition overall?
2. What are the average goals scored for a team overall?
3. What are the average goals conceded in a game for a team overall?
    a. Visualize Win percentages of Top 10 teams according to win percentage against teams with varying win percentages by average goals scored and conceded.
    b. Visualize Win percentages of Top 10 teams according to win percentage against teams with varying win percentages by average goals conceded.
4. We now can use our model to get the coefficient! Coefficients are specifically weighted by the previous factors calculated, using a simple linear equation
5. Perform Machine Learning using a classifier() model using coefficient, and a classifier() model without using the coefficient to predict the winner of the UEFA Champions League and evaluate models' accuracy of prediction.

Motivation and Background

The UEFA Champions League is one of the most popular sports tournaments worldwide. Certainly many people gamble and bet on who will win, while fans want to know what hope may hide beneath the advanced analytics that may show how much of a chance their team has in the competition.

In our project, we hope to uncover and reinterpret what the scores of previous matches tells us about how well a team will perform against opponents they've never faced simply based on their

performance against teams of varying win percentages. This data would also be helpful to the teams themselves. They can see which teams are the ones that seem like easy wins, but hidden in their record against teams that score a similar number of goals is the reality that those wins aren't too easy to come by.

In the betting industry, being able to predict the winner of a game is paramount in the house winning most of its earnings and determining odds for probable events. Thus, this project will serve to provide quantifiable answers to that real world need, by wrangling, aggregating and creating coefficients from the data and using machine learning to predict the future winner(s) of the Champions League.

Dataset

The Dataset can be found at this URL:
https://github.com/jalapic/engsoccerdata/blob/master/data-raw/champs.csv

This is a dataset that contains data on the scores of UEFA Champions League Matches from 1955 to 2015, including finals data from 2016 and 2017. The data describes the entire champions league competition for that season, indicating which level of the competition (semi-finals, quarter-finals, etc) the game took place in. We will subset this data to exclude qualifier match data and contain data beginning from the group stages of the competition for the years 2008-2015.

Since this dataset is on GitHub and open to the public, we will be using the *BeautifulSoup* and *Requests* libraries to scrape the data if it is not found in the correct directory.

Methodology

The analysis is structured in such a way *that it leads to the machine learning problem*.

We follow the steps as presented below:

1. Subset the existing dataset to get the data from 2008-2015 and excluding qualifier matches. The year 1992 is chosen as that is when the UEFA Champions League took its current form.
2. Group the Dataset by team, and using the 'tiewinner' field, we can count the number of times the team's name appears on it
3. Group the dataset by team, and get the get the number of games played by counting all games played by the team

4. Divide the number of wins of the team by the number of games played by the team and keep it in a new column
5. Print out the top 10 win percentages, so we have an idea of the probable teams to win the Champions League.

Now, we move on to Average goals and Actual goals Scored.
All of this is done for when the team is home, and when the team is away. So we will group the dataset by team when the team is at home, and when the team is away for steps 7 through 12.

6. Group the dataset by team.
7. Calculate the sum of goals scored when the team is the home team by summing the *'hgoal'* column, and when the team is away by summing the *'vgoal'* column.
8. Add both the number of goals scored together to get the actual goals scored by the team.
9. We divide the value of actual goals by team by the number of matches each team plays, to get the average goals scored.
10. Since we already grouped the data by team, home and away, we can also get the goals conceded and average goals conceded by summing up the *'vgoal'* column when the team is home, and the *'hgoal*' column when the team is away.
11. We then add the two conceded goals columns to get the total number of goals conceded. We divide that by the total number of matches played by the team, using the same algorithm as in (3.) to get the average goals conceded by a team.

Knowing the average goals scored and conceded in a game by a team, we can now quantify whether the team has a leaky defence and/or a misfiring attack- something that teams will benefit from knowing.

12. Visualise the Teams by win percentages, with a size parameter on the plot for the average goals scored.
13. Visualize the Teams by win percentages, with a size parameter on the plot for average goals conceded.

Through these Visualizations, we can see the impact the goals scored, and more importantly, the goals conceded, have on the team's ability to get a positive result, as shown by the win percentage.

14. Compile the columns of win percentages, average goals and number of goals scored and conceded by a team(home and away in separate columns) into a single dataframe by joining the aggregated dataframes.

Keeping all of the relevant data in a new dataframe makes it easier to perform machine learning, while also having the added advantage of having all your information in one place.

Now we can move on to creating the coefficient. The Coefficient is a numeric value based on a simple linear equation whose variables are the numeric values in the dataframe.

We will weight the following variables as follows:

1. Win Percentage = 0.4(40%)
2. Average Goals Scored(Home) = 0.08(8%)
3. Average Goals Conceded(Home) = 0.22(22%)
4. Average Goals Scored(Away) = 0.22(22%)
5. Average Goals Conceded(Away) = 0.08(8%)

We chose to weight it as such, because of the introduction of the away goals rule in the Champions League, which favors goals scored by teams away from home in case of a tie in aggregate goals scored by the teams in the fixture

Thus, the coefficient will be calculated as:

$$q = (0.4 \times Win_{\%}) + (0.08 \times AvgScored_{home}) - (0.08 \times AvgConceded_{away}) + (0.22 \times AvgScored_{away}) - (0.22 \times AvgConceded_{home})$$

Where q is the coefficient.

Now, we perform machine learning, using a classifier model with a test size of 20% on finding the victor based on the coefficient values, using which we will know using past results, which team is projected to win the Champions League.
We will also use another classifier model without the use of a coefficient and with the same test size, to predict the winner of the Champions League.
This is done so that the inbuilt machine learning algorithm can predict using the raw data of the win percentage, average goals scored and conceded, and we can see whether our highest calculated coefficient was actually that of the actual winner.
We will compare our predicted result with the actual winners of 2016, 2017 and 2018 to evaluate both our model's accuracy in predicting the winner, based on the values that we calculated and the coefficient of the predicted winner, from the curated dataframe.

Work Plan

In order to collaborate on our code, we will push all our code to a private repository on GitHub that we all can access at any time. Whenever one of us makes a change to the code, we will push our changes immediately.

For every task below, the team member responsible will also be responsible in documenting this part of the project in the final report. They will also be responsible for documenting the part of the code they write.

Task	**Person Responsible**
Write code to parse GitHub page with dataset, filter out extraneous characters and turn into DataFrame/csv file in the correct directory for the script to run.	Alex Eidt
Analyze the dataset using the procedure shown above	Alex Eidt: Steps 1-7 Pranav Natarajan: Steps 8-15
Use *seaborn* and *matplotlib* to plot the data from our new DataFrame with values we calculated from the second task.	Alex Eidt
Use *sklearn* and classifier() models to predict the victor based on the weighted coefficient values, and using raw data in the dataset	Pranav Natarajan
Review test cases and make sure all code is well documented.	Pranav Natarajan
Revise the final draft of the report to make it coherent and correct any mistakes.	Alex Eidt